

(19) **United States**(12) **Patent Application Publication**  
**SAETHER et al.**(10) **Pub. No.: US 2001/0042073 A1**(43) **Pub. Date: Nov. 15, 2001**(54) **METHOD AND SYSTEM FOR  
AUTOMATICALLY UPDATING THE  
VERSION OF A SET OF FILES STORED ON  
CONTENT SERVERS**(76) Inventors: **CHRISTIAN D. SAETHER,**  
SEATTLE, WA (US); **DAVID E.**  
SLOAT, SEATTLE, WA (US)

Correspondence Address:

**MERCHANT & GOULD PC****P.O. BOX 2903****MINNEAPOLIS, MN 55402-0903 (US)**

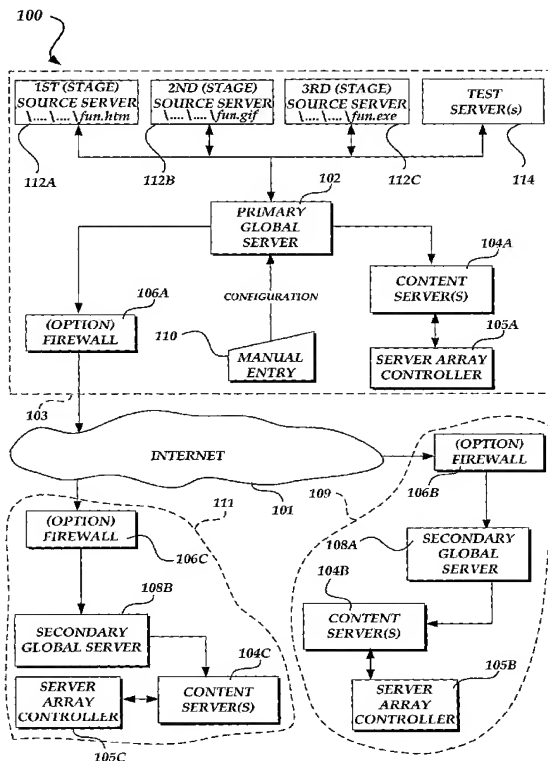
( \* ) Notice: This is a publication of a continued prosecution application (CPA) filed under 37 CFR 1.53(d).

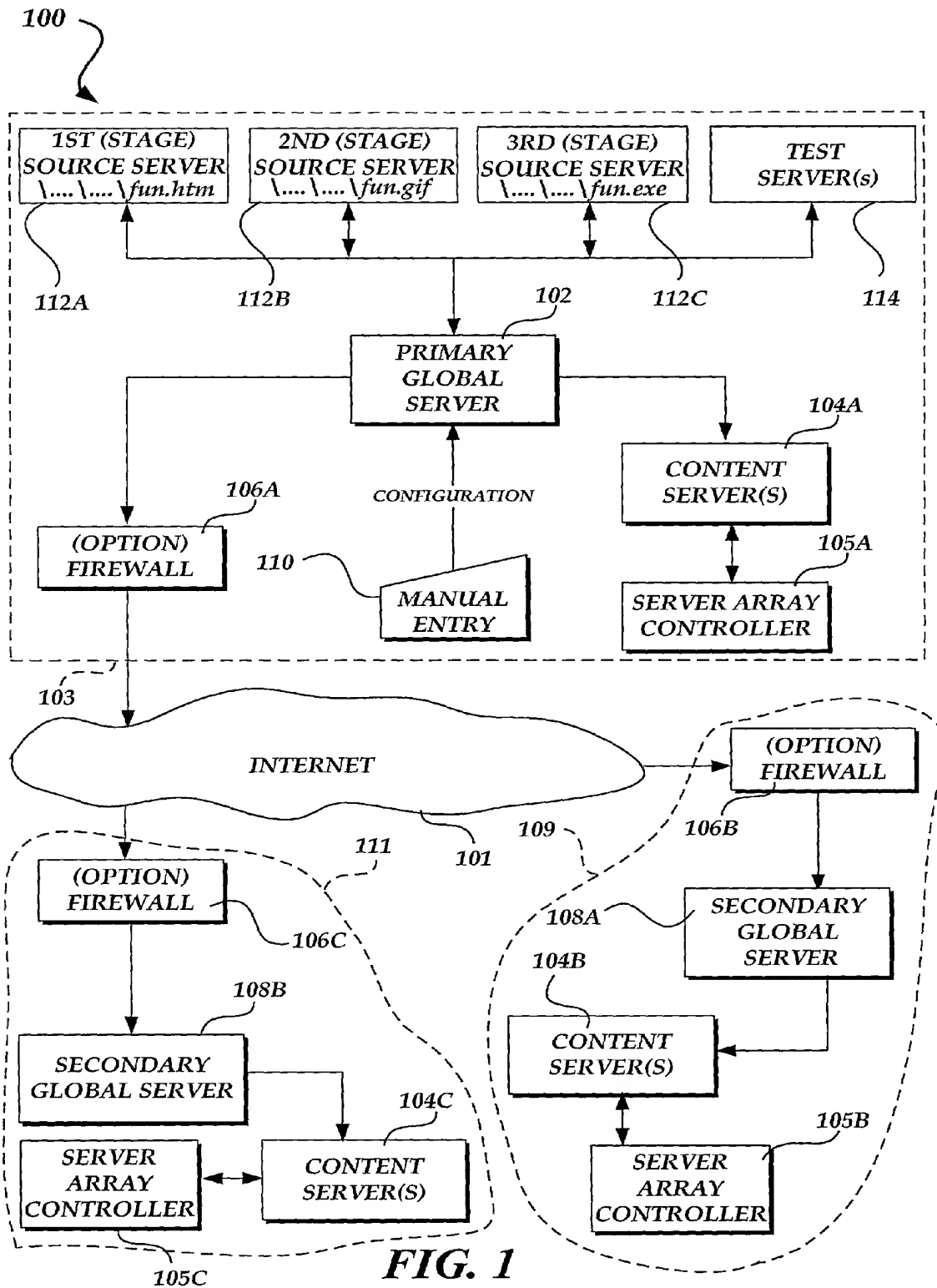
(21) Appl. No.: **09/405,894**(22) Filed: **Sep. 24, 1999****Related U.S. Application Data**

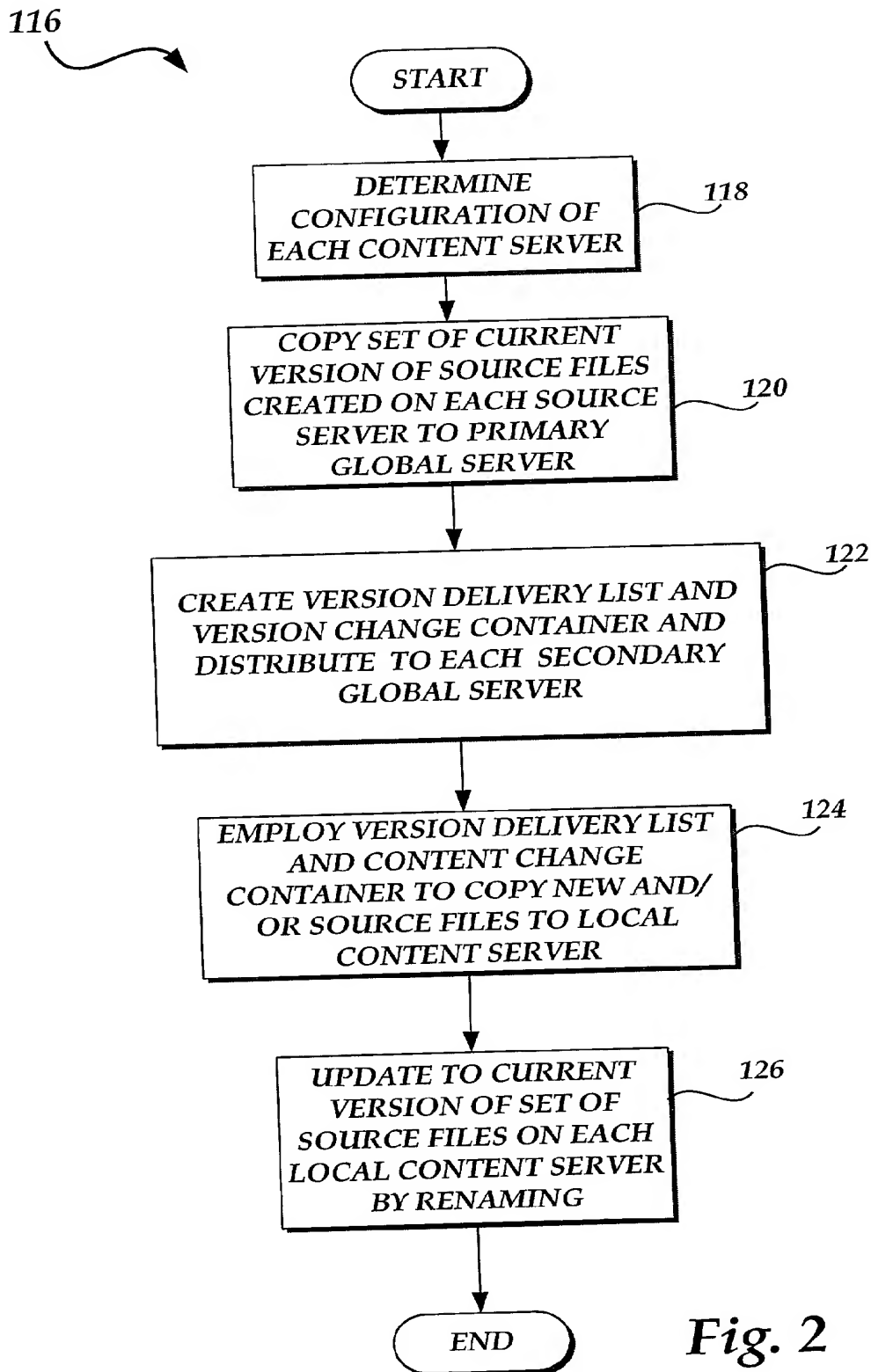
(63) Non-provisional of provisional application No. 60/140,213, filed on Jun. 22, 1999.

**Publication Classification**(51) **Int. Cl.<sup>7</sup> ..... G06F 17/30; G06F 15/16**(52) **U.S. Cl. .... 707/203; 709/219; 707/10;  
707/200**(57) **ABSTRACT**

A method and system for managing the replication and version synchronization of updates to a set of source files on geographically distributed heterogeneous content servers with minimal impact on a network's bandwidth. The configuration of each content server is either manually entered or automatically determined. The current version of the source files are created on at least one source server. A Primary global server stores a copy of the current version of the set of the source files along with the configuration of each content server. The Primary global server generates and distributes a particular version change container and version distribution list to each remotely located Secondary global server. Each Secondary global server employs the version distribution list and the contents of the version change container to identify the current version of each source file necessary to upgrade the set of source files on each local content server. Each identified source file is copied to a sub-directory on each local content server associated with the Secondary global server. At each local content server, the renaming of each copied source file is employed to update to the current version of the set of source files on the content server. A versioned file tree repository for the set of source files includes archived objects. When the version distribution list identifies a previous version, the current version of source files on the local content servers can be rolled back to the previous version.







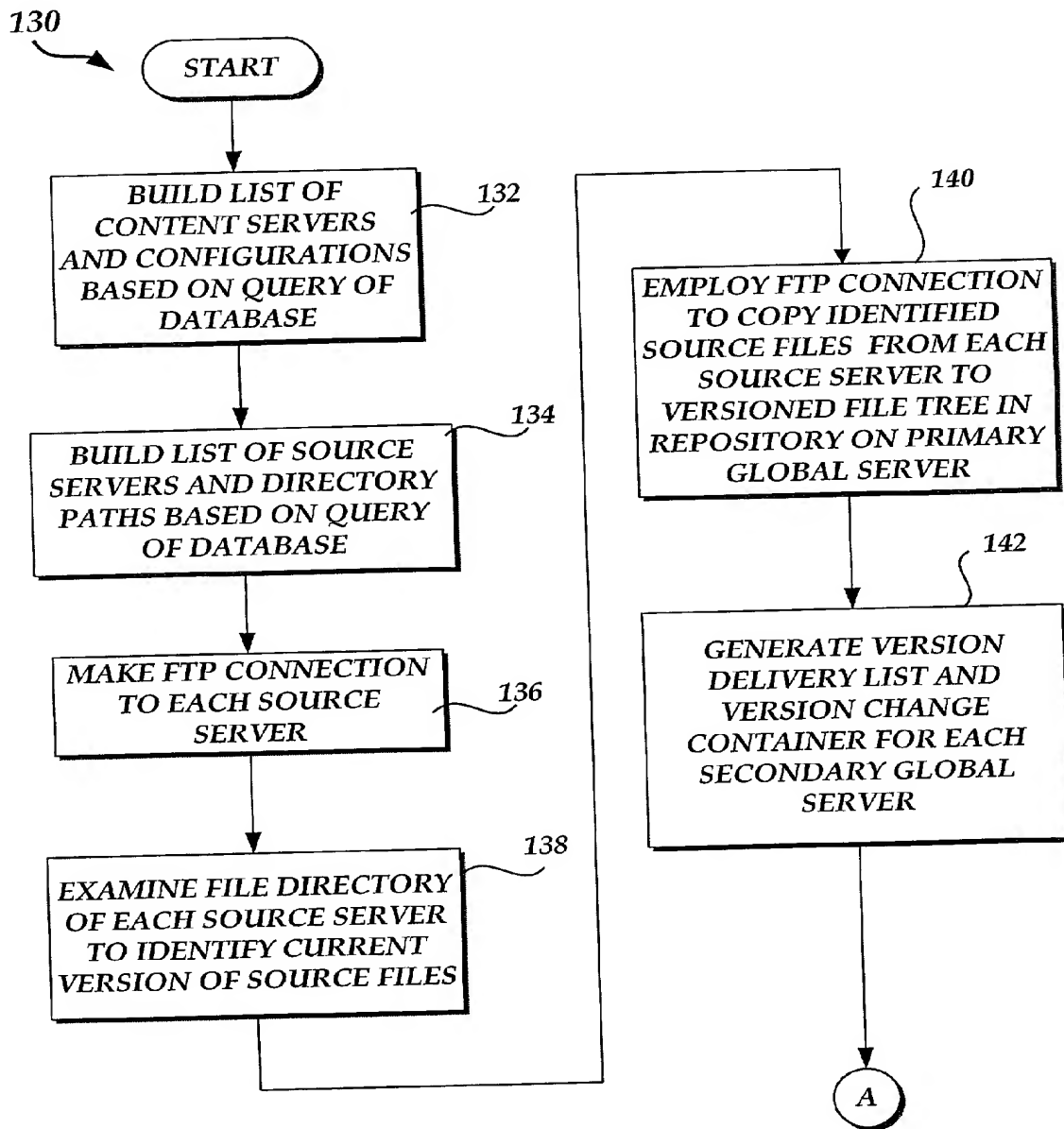


Fig. 3A

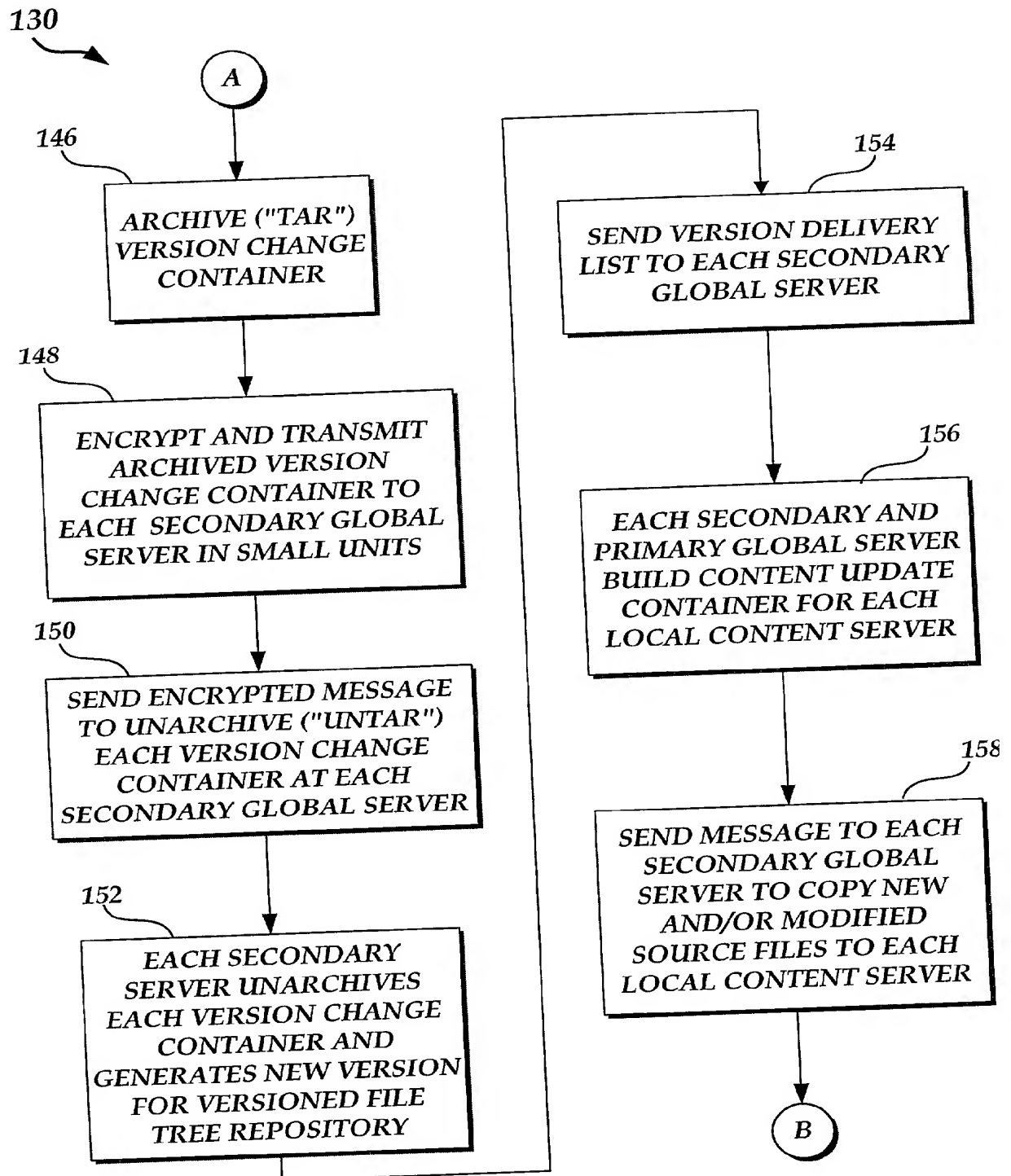


Fig. 3B

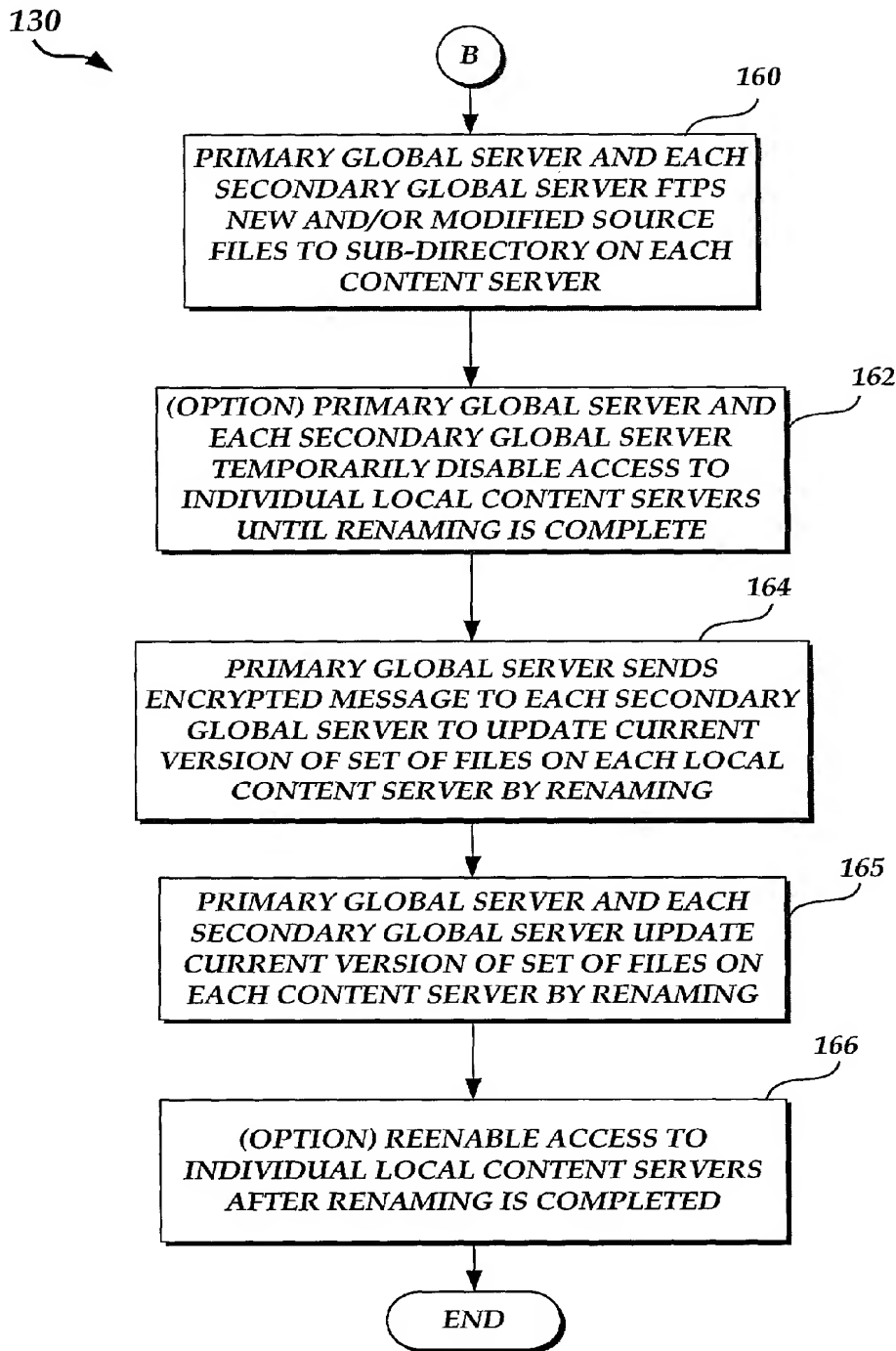
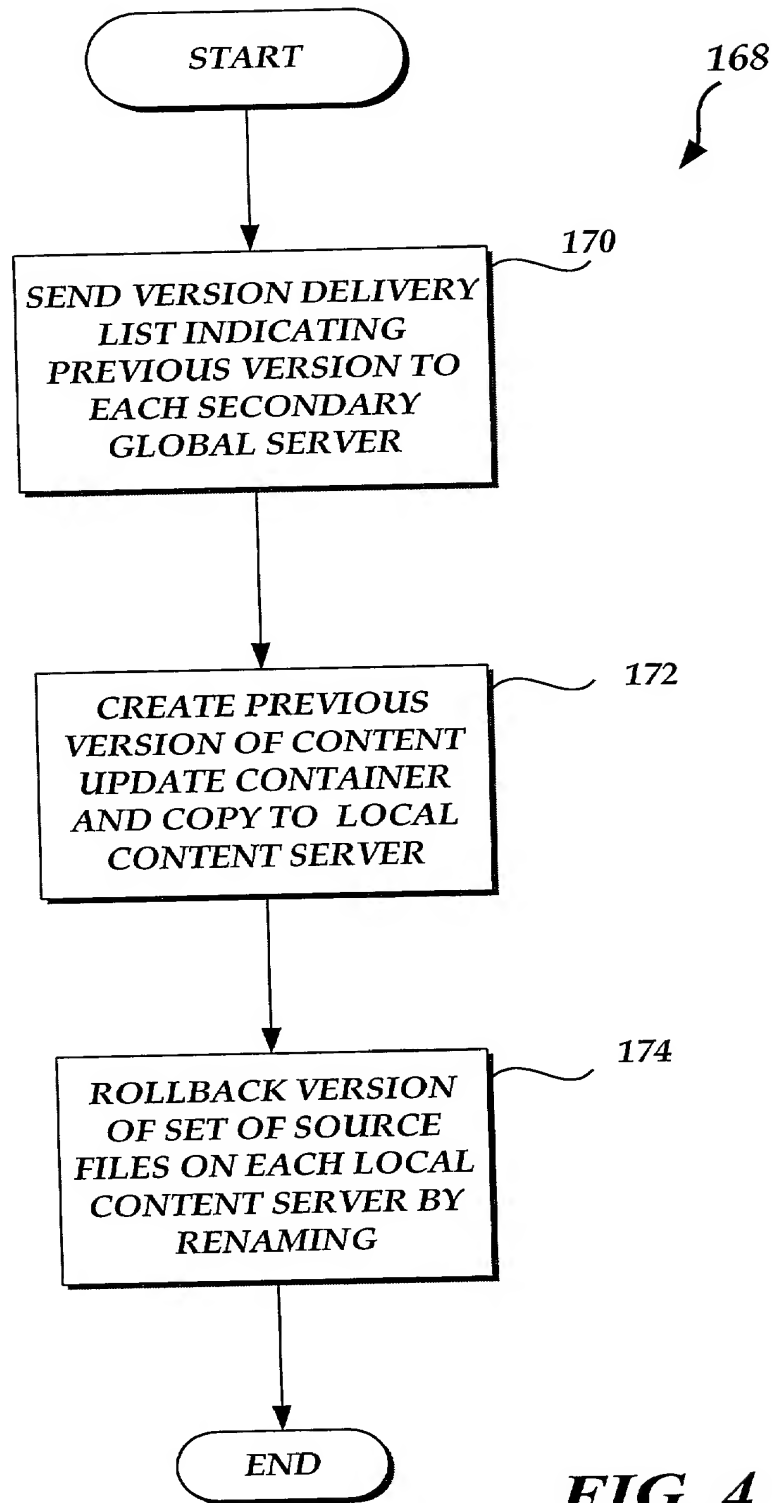
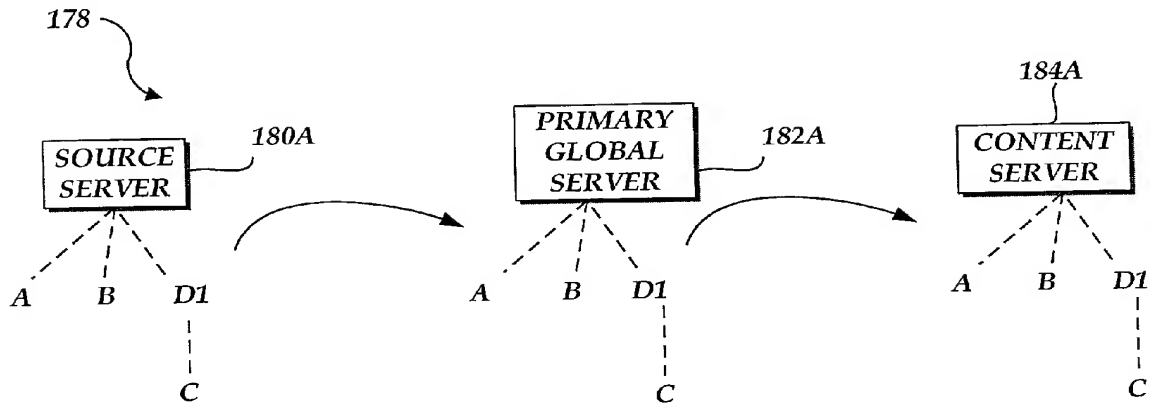


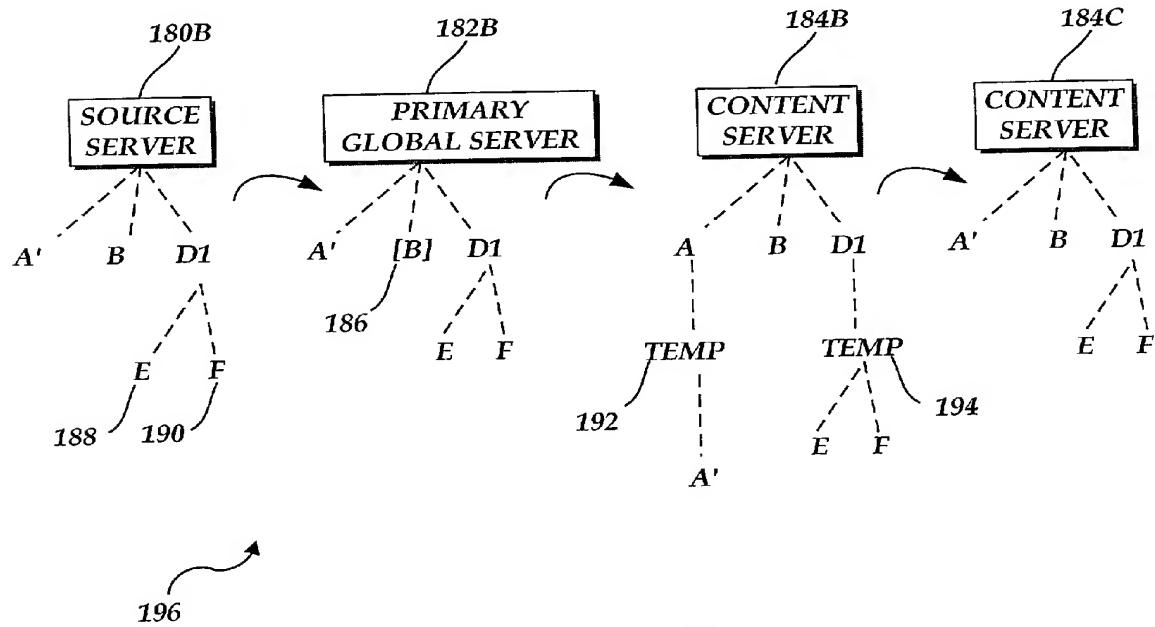
Fig. 3C



**FIG. 4**

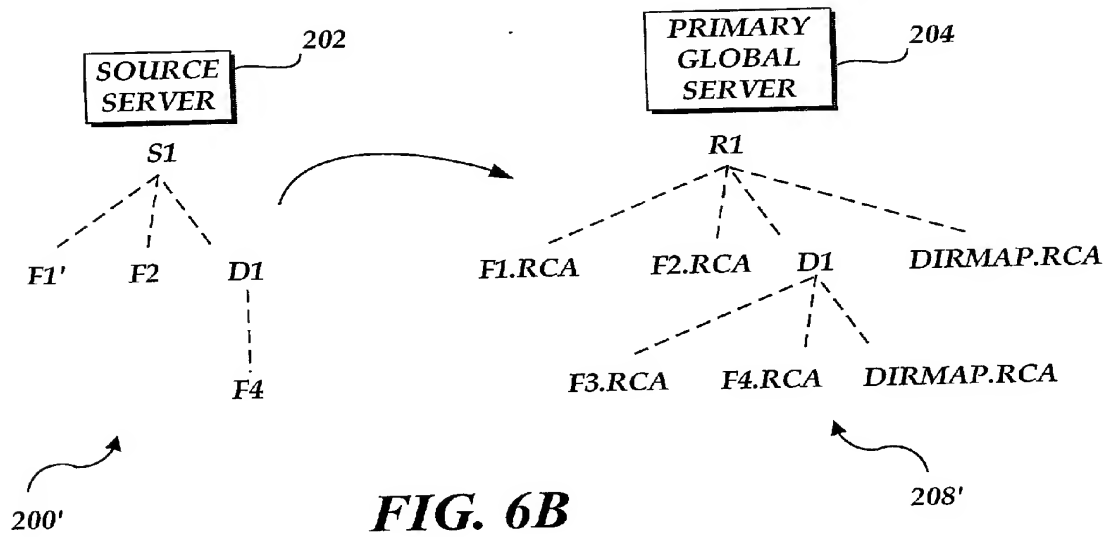
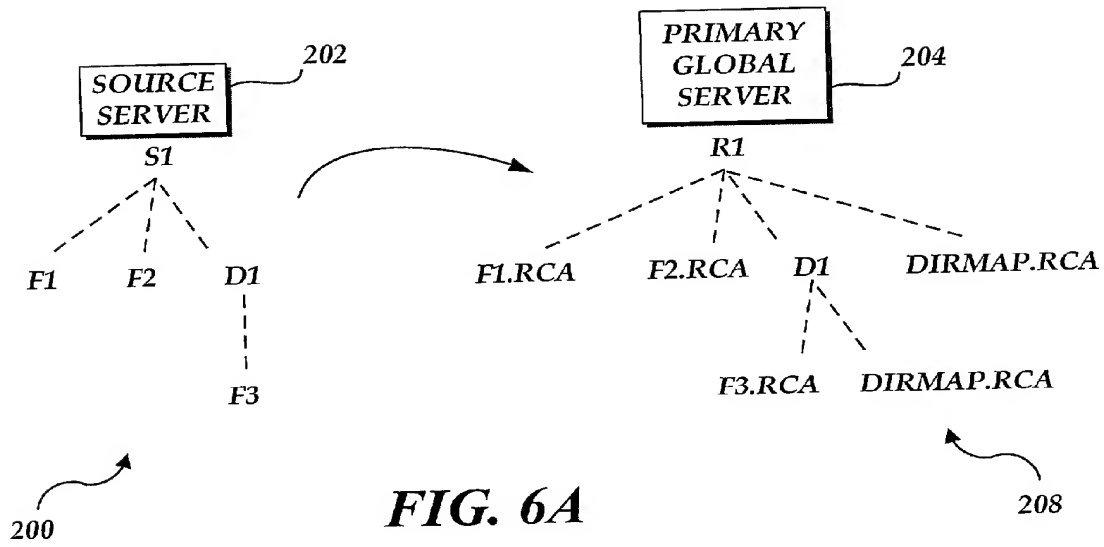


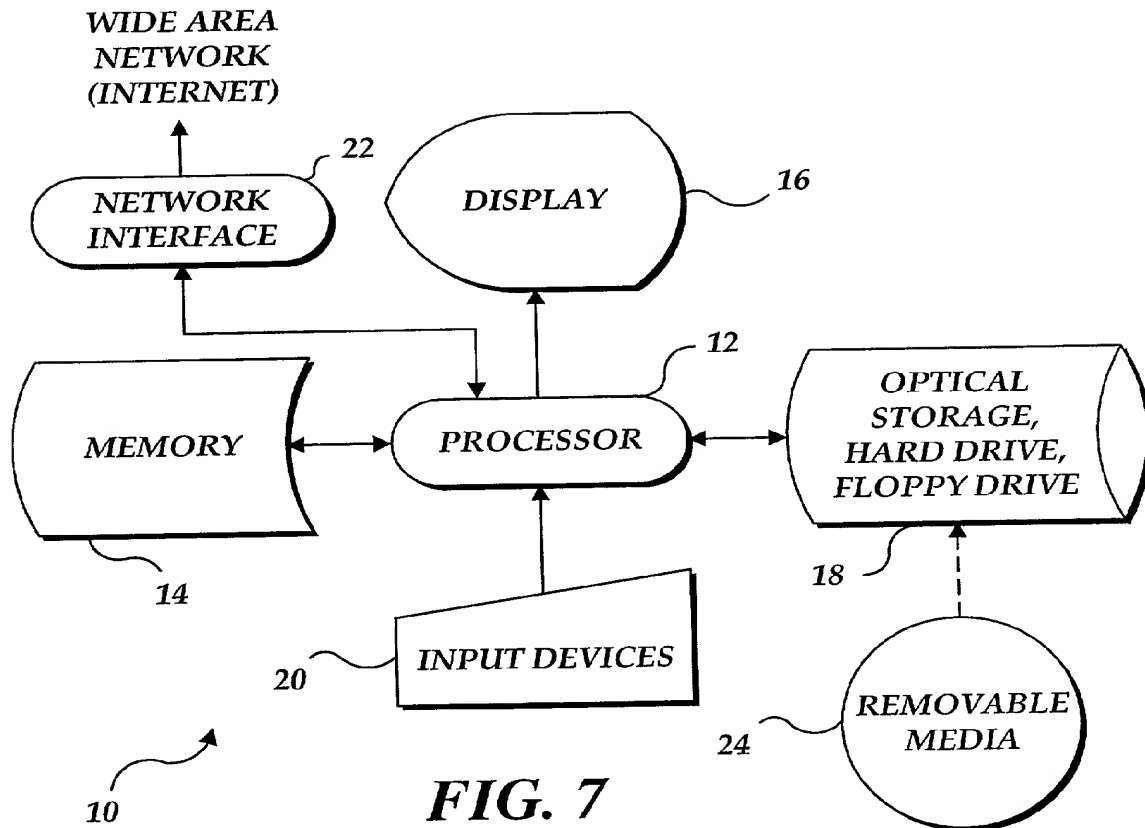
**FIG. 5A**



**FIG. 5B**







## METHOD AND SYSTEM FOR AUTOMATICALLY UPDATING THE VERSION OF A SET OF FILES STORED ON CONTENT SERVERS

### FIELD OF THE INVENTION

[0001] This application relates generally to distributing updates to geographically distributed servers on a network, and, more specifically, to enabling the version of each source file stored on heterogeneous content servers to be automatically updated.

### BACKGROUND OF THE INVENTION

[0002] Often, source files for web content servers are coded by multiple programmers on remotely located (stage) source servers. It is not unusual for one programmer(s) to code "HTML" files on one source server while another programmer(s) creates executable and/or image files on another source server. Once a programmer debugs a newly created/edited update file, it is eventually distributed to each content server and placed in a corresponding file directory. Historically, the distribution of the current version of a set of "updated" or new files from remotely located source servers through the Internet to content servers has proven to be a difficult task for several reasons. One reason is that the file directory structure and hardware configuration can vary between individual web content servers. In this case, the distribution of a set of files for each web content server must be separately organized according to each server's file directory structure and hardware capabilities. Another reason is that the actual size of the set of files may be so large that their distribution is relatively slow on a network with limited bandwidth capabilities.

[0003] Therefore, a need exists for a computer implementable method of distributing a set of the current version of source files to a plurality of content servers using a minimal amount of bandwidth. Preferably, the method will tailor the distribution of the set of source files according to the configuration, i.e., file structure and the hardware constraints, of each content server. Also, preferably the method would provide a facility for rolling back the current version of the set of source files to a previous version. The present invention is directed to providing such a computer implementable method.

### SUMMARY OF THE INVENTION

[0004] In accordance with the present invention, a computer implementable method for updating a version of a set of source files stored on a content server over a network, comprising: (a) determining a configuration of each content server on the network, the configuration enabling a source file to be copied to a location on the content server; (b) identifying each source file on a source server that is different than any source file stored on a global server; (c) copying each identifiably different source file from the source server to the global server, each source file copied from the source server and a set of source files stored on the global server being employed to create a current version of the set of source files on the global server; and (d) employing the configuration of each content server to copy the current version of each source file that is included in the set of source files on the global server to a directory created on each content server, whereby the version of the set of source

files stored on each content server is updated by renaming the current version of each source file copied to the directory on each content server.

[0005] In accordance with other aspects of the present invention, the method provides for renaming each current version of each source file that is copied to the directory created on each content server; and deleting the directory created on the content server and deleting another version of each source file that is updated by the renaming of the current version of each source file copied to the content server directory.

[0006] In accordance with yet other aspects of the present invention, the method provides for when the current version of each source file is copied to the directory created on each content server, disabling access to the set of source files on a particular content server until the renaming of the current version of each source file copied to the directory on the particular content is completed.

[0007] In accordance with still other aspects of the present invention, the method provides for when the current version of each source file is copied to the directory created on each content server, starting the renaming process with the current version of each copied source file that is furthest away from the root directory of each content server.

[0008] In accordance with other aspects of the present invention, the method provides for archiving each version of the set of source files in a repository on the global server, the archiving causing each source file to be individually compressed and stored as an archived object in the repository associated with the global server. The repository can be a versioned file tree repository for the set of source files.

[0009] In accordance with still further aspects of the present invention, the method provides for when a return to a previous version of the set of source files is requested, retrieving each archived object associated with the previous version of the set of source files from the repository associated with the global server. Each archived object associated with the previous version of the set of source files is unarchived to reconstitute each source file needed to upgrade the set of source files on the content server to the previous version. Each reconstituted source file is copied to a directory created on each content server, whereby the version of the set of source files on each content server is upgraded to the previous version by renaming the copied reconstituted source files.

[0010] In accordance with still other aspects of the present invention, the method provides for enabling a user to edit the configuration for each content server. Alternatively, the method may provide for automatically obtain the configuration for each content server.

[0011] In accordance with other aspects of the present invention, the method provides for employing a file access protocol to gain file level access to each source file, including FTP, NFS, CIFS and MFTP. The file access protocol may employ one port to send and receive data that includes a message and a source file. The type of source file includes image, hyper text mark-up language (IITML), script, sound, video, text, picture and application program code.

[0012] In accordance with yet other aspects of the present invention, the method provides for when a new content

server is added to the network, employing the current version of the set of source files stored in a repository on the global server and a configuration of the new content server to replicate the current version of the set of source files in at least one directory created on the new content server.

[0013] In accordance with still further aspects of the present invention, the method provides for copying the differences in the set of source files on the source server to a primary global server which generates a particular container that includes the differences in the set of source files stored on each remotely located secondary global server. The primary global server distributes the particular container from the primary global server to each associated secondary global server which employ the contents of the particular container to replicate the current version of the set of source files in a repository on the Secondary global server. The current version of each source file stored in the repository on the Secondary global server that is identified as necessary to replicate the current version of the set of source files on the content server is copied to another directory created on each content server that is local to the secondary global server. The set of source files may be stored in a versioned file tree repository on the primary global server and each secondary global server,

[0014] In accordance with yet other aspects of the present invention, the method provides for automatically distributing the container to the secondary global server. Alternatively, the distribution of the container to the secondary global server can be selectively enabled by an input. Also, the updating to the current version of the set of source files on the content server can be automatic or selectively enabled by an input. Additionally, each container can be distributed

computer-executable instructions that may be used to perform substantially the same methods as those described above is provided.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0020] The foregoing aspects and many of the attendant advantages of this invention will become more readily appreciated as the same become better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings, wherein:

[0021] FIG. 1 illustrates an overview of the system architecture for implementing the present invention;

[0022] FIG. 2 is a flowchart showing an overview of the logic for updating files on remotely located content servers;

[0023] FIGS. 3A-3C are flowcharts that illustrate in more detail the logic for updating files on remotely located content servers;

[0024] FIG. 4 is a flowchart showing the logic for rolling back a version of files on remotely located content servers;

[0025] FIG. 5A is an overview of the file directory structure for an initial version of a set of source files that are created on a source server and copied to a Primary global server and a content server;

[0026] FIG. 5B is an overview of the file directory structure for an updated version of the set of source files that are created on the source server and copied to the Primary global server and the content server;

[0027] FIG. 6A is an overview of the initial versioning of a source tree that is created on the source server and copied